# Ridge Regression Example

Nirupama Tamvada

### Model Fitting Algorithm

The `mtool` implementation uses stochastic gradient descent, and thus is well-suited in terms of speed to problems with a large `N`.

We simulate data from a multinomial distribution with 3 classes (class 0 is the reference class). To return the correct results from `mtool`, it is very important to make sure that `Y` is numeric and that the categories are coded as 0, 1, 2, …etc. with 0 as the reference.

### Case : Sparse settings with regularization ($\alpha = 0.5$, Ridge Regression)

### Elastic Net Penalization in `mtool`

The optimization performed on `mtool` calls the `Proximal Toolbox` , which is a toolbox in the larger `SPAMS` (SPArse Modeling Software) optimization toolbox for sparse decomposition problems. For the Elastic-Net penalty, for every column of $u$ of $U = [u^1, ..., u^n]$ in $\mathbb{R}^{p\times n}$, one column of $V = [v^1, ..., v^n]$ in $\mathbb{R}^{p\times n}$ is computed to solve the following proximal operator:

$$\min_{v\in\mathbb{R}^p} \frac{1}{2}||u - v||_2^2 + \lambda_1||v||_1 + \lambda_2||v||_2^2$$

We set $\lambda_1 = \lambda\alpha$ and $\lambda_2 = \frac{\lambda(1-\alpha)}{2}$ to obtain the Elastic Net penalty implemented in `glmnet` as well from Zou and Hastie., 2005.

### 1. N > p (N = 1000, p = 10)

```r
set.seed(200)
# Generate covariates
X <- matrix(rnorm(10000), 1000, 10)
```

```r
# coefficients for each choice with some sparsity
X1 <- rep(0, 10)
X2 <- c(rep(0.5, 5), rep(0, 5))
zero_X2 <- which(X2 == 0)

X3 <- c(rep(-1, 4), rep(0, 6))
zero_X3 <- which(X3 == 0)


# vector of probabilities
vProb = cbind(exp(X%*%X1), exp(X%*%X2), exp(X%*%X3))

# multinomial draws
mChoices <- t(apply(vProb, 1, rmultinom, n = 1, size = 1))
dfM <- cbind.data.frame(y = apply(mChoices, 1, function(x) which(x == 1)), X)
# Rename covariates
colnames(dfM)[2:11] <- paste0('x', 1:10)

# 0, 1, 2 for levels
Y <- as.numeric(dfM$y-1)

# Covariate matrix
X <- as.matrix(dfM[, c(2:11)])

# Rename covariates
colnames(X) <- paste0('x', 1:10)
```

Fitting the two models (`nnet` does not provide a penalized implementation of the multinomial regression):

```r
# glmnet
  fit.glmnet <- fit.glmnet <- glmnet::glmnet(
    x = X, y = Y,
    family = "multinomial",
    intercept = FALSE,
    type.multinomial = "grouped",  # same sparsity pattern for all outcome classes
    lambda = 0.1, alpha = 0)
   # Elastic-net reparametrization
  alpha <- 0
  lambda <- 0.1
  lambda1 <- lambda*alpha
```

```
    lambda2 <- 0.5*lambda*(1 - alpha)
    # mtool
    fit.mtool <- mtool::mtool.MNlogistic(
  X = cbind(X),
 Y = Y,
 offset = rep(0, length(Y)),
     N_covariates = 0,
     regularization = 'elastic-net',
     transpose = FALSE,
     lambda1 = lambda1, lambda2 = lambda2,
               lambda3 = 0
 )
```

Results table for coefficients for class 2:

| Covariates | True coefficients | glmnet | mtool |
|---|---|---|---|
| x1 | 0.5 | 0.392 | NaN |
| x2 | 0.5 | 0.432 | NaN |
| x3 | 0.5 | 0.391 | NaN |
| x4 | 0.5 | 0.451 | NaN |
| x5 | 0.5 | 0.129 | NaN |
| x1 | 0 | -0.039 | NaN |
| x2 | 0 | -0.086 | NaN |
| x3 | 0 | -0.021 | NaN |
| x4 | 0 | 0.026 | NaN |
| x5 | 0 | 0.034 | NaN |